

Variance based Scheduling to Improve the QoS Performance at the Cell Edge

Trshant Bhat and Sanjay Singh*

August 16, 2012

Abstract

Now a days mobile phones are most often used for data communication rather than voice calls. Due to this change in user behavior, there is a need to improve the QoS received by the user. One of the ways of improving the QoS is an efficient scheduling algorithm which incorporates the needs of the users and variation in channel condition. The parameters used to measure the efficiency of the scheduling algorithms are the Jain Fairness Index and the overall system throughput. In this paper we have proposed a variance based scheduling algorithm which selects the user who has the highest variance of data transmitted in a given time frame as a parameter for scheduling. This ensures that eventually, the users transmit almost equal amounts of data regardless of channel condition. The simulation results shows that the proposed algorithm achieves high Jain Fairness Index of 0.92 with a lesser drop in the system throughput 18% as compared to Dynamically altering Proportionally Fair Algorithm's 20% using the Proportionally Fair Algorithm as reference.

1 Introduction

Mobile phones were first introduced with the view of having usability of telephones but with the convenience of mobility. The infrastructure is primarily based on radio communication but multiple users are simultaneously supported by slotting them by frequency or time to improve the number of users supported. This method has limitations in transmission but since the human ear is resilient to slight errors in transmission of voice, this was considered enough originally.

As time passed by and there was innovation in mobile Internet technologies (i.e., GPRS, 3G, 4G) accompanied by improvements in compression technologies (mp3, flv); the use of mobile phones expanded to much more than just telephony. However, data transmission now had to be error free, or in its absence, the user would have to transmit more data to replace the erroneously

*Sanjay Singh is with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal University, Manipal-576104, INDIA, E-mail: sanjay.singh@manipal.edu

transmitted data. The infrastructure is split into cells for better management, but due to propagation losses and multipath fading, as the distance of the user from the cell increased, the capacity of the user to transmit data decreased. While the effect of this is negligible in voice communication, this negatively affected data transmission for users on the periphery of the cell. The advances in transmission technologies has decreased the transmission errors, but there is a need for improved scheduling algorithms that would satisfy all the users in all parts of the cell.

Proportionally Fair Algorithm (PFA) [1] tries to solve this by using the concept of proportional fairness to schedule the users, but it eventually schedules all the users equally, causing the users in different parts of the cell to send varying amounts of data. Dynamically Updating Proportionally Fair Algorithm (DPFA) [2] tried to increase the throughput of the users on the cell edge by restraining the users at the cell center, but this created unfairness to the users at cell center, along with a drop in the overall system throughput.

The primary objective of this paper is to demonstrate the use of variance in addressing the limitations of the scheduling algorithms as described above.

Rest of the paper is organized as follows. Section 2 explains the evaluation parameters used for comparing the efficacy of a scheduler. Section 3 explains the existing scheduling algorithms PFA and DPFA. Section 4 describes the proposed algorithm. Section 5 discusses the simulation results and finally section 6 concludes the paper.

2 Scheduling Algorithm Evaluation Parameters

Parameters used to evaluate the efficacy of scheduling algorithms are:

1. Overall System Throughput
2. Fairness Index

The system throughputs is a counter maintaining account of the overall system throughput.

Fairness Index

The concept of fairness [3] in mobile systems arises when a limited amount of bandwidth is to be shared among several users simultaneously. It is highly desirable to find a method and, preferably, an index to measure and compare the degree of fairness of a particular allocation policy. This index is called the Fairness Index (FI). The Fairness Index and throughput maximization are two main performance metrics that are often used to analyze and compare the performance of different resource allocation schemes, since these two metrics define the service provider and consumers satisfaction index respectively.

Jain fairness index [4] is given by,

$$I_{Jain} = \frac{|\sum_{i=1}^n \bar{r}_i|^2}{N \sum_{i=1}^n \bar{r}_i^2} \quad (1)$$

where \bar{r}_i is the throughput of the i th users.

The Jain Fairness index is used to measure the FI because of its computational simplicity. As per the definition of fairness index [4], the system would be fairest when the throughput of the users would be equal.

3 Existing Scheduling Algorithms

In this section, we describe and discuss two of the existing scheduling algorithms PFA and DPFA respectively.

3.1 Proportionally Fair Algorithm

In the Proportionally Fair Algorithm (PFA) [1], the selection of the user at any time is based on a balance between the current possible rates and fairness. This algorithm performs scheduling by calculating the ratio of given rate for each user with their average throughput to date, and selecting the one with the maximum ratio, that is

$$P_{k(PFA)} = \arg \max_k \frac{r_k}{R_k} \quad (2)$$

where r_k , in the numerator represents the instantaneous data rate of the k th users while R_k represents the average rate of the same user based on the past resource allocation. The PFA allocates the resource at a given time to the user who has the maximum value of $P_{k(PFA)}$. The average throughput R_k is defined [1] as,

$$R_k[t] = \begin{cases} (1 - \frac{1}{T_c})R_k[t-1] + \frac{1}{T_c}r_k[t], & \text{if } k \text{ is scheduled} \\ (1 - \frac{1}{T_c})R_k[t-1], & \text{if } k \text{ is not scheduled} \end{cases} \quad (3)$$

where T_c is the system time count which is initialized to 0 and increments every time slot.

The rate R_k is updated every time slot after choosing the user. The rate updating equations are observed to update such that the average rate of the chosen user drops drastically while the unchosen user drops a little. As a result, in the next round of selection, the priority of the unchosen user increases exponentially whereas the priority of the chosen user increases marginally.

The PF scheduling is simple but effective for non real-time services. However it has been observed that this algorithm schedules all the users equally, hence the user at the cell edge would not be able to transfer as much data as the user at the cell center due to the degraded channel condition. This algorithm starves the users at the cell edge. Hence degradation in QoS performance. There is also a disparity in the individual user throughput.

3.2 Dynamically Updating Proportionally Fair Algorithm

Ning et al [2] has proposed the modification to PFA. The drawbacks of the PFA are remedied by using two controlling factors α and β respectively, which is used as,

$$P_{k(DPFA)} = \arg \max_k \frac{(r_k)^\alpha}{(R_k)^\beta} \quad (4)$$

where ,

if $\alpha = \beta = 1$, then we get PF scheduling

if $\alpha = 1, \beta = 0$, then it is a maximum C/I scheduling [5]

if $\alpha = 0, \beta = 1$, then it is a round-robin (RR) scheduling [6]

The value of α remain a small positive value usually 1, and the β value will change for every user. There are two timers introduced in this algorithm, $A_k[t]$ and $B_k[t]$ to record the duration in which the user remain in the cell edge and at the cell center, respectively. $A_k[t]$ and $B_k[t]$ have been defined as,

$$A_k[t] = \begin{cases} 0, & \gamma_k[t] \geq \delta \\ A_k[t-1] + 1, & \gamma_k[t] < \delta \end{cases} \quad (5)$$

and

$$B_k[t] = \begin{cases} 0, & \gamma_k[t] > \delta \\ B_k[t-1] + 1, & \gamma_k[t] \leq \delta \end{cases} \quad (6)$$

where δ is the cell SNR threshold and $\gamma_k[t]$ is the SNR of the k th user. Hence with these two equations, β is updated as

$$\beta = \begin{cases} 1, & A_k[t] \geq \theta \text{ or } 0 \leq B_k[t] \leq \theta \\ \max\{\frac{\gamma_k[t]}{\delta}, b\}, & 0 \leq A_k[t] \leq \theta \text{ or } B_k[t] > \theta \end{cases} \quad (7)$$

for all k users. Parameter θ is the cell time threshold and b is a constant usually equal to 0.5. As per equation 7, the users in the cell edge are compensated and the users in the cell center are punished. Since the users at the cell edge are given higher priority in this algorithm, the throughput of those users is increased at the expense of the users at the center. Due to the fact that the users at the edge are scheduled more often, whose datarate is low as compared to users at cell center, the overall system throughput decreases as compared to the pure PF algorithm. The disparity in the individual user throughput reduces, increasing the FI to 0.86.

4 Proposed Scheduling Algorithm

Following points has been considered while formulating the algorithm:

1. As the distance from the tower increases the user is capable of sending lesser data.
2. Punishing and compensating should be kept to a bare minimum.

3. It should be computationally simple.

In rest of the paper, we will call the proposed algorithm as VPFA (Variance based Proportionally Fair Algorithm). We first define two counters,

1. **Individual User Throughput:** This counts the throughput of the individual user.
2. **FI Stability Counter (C_s):** This counts the evaluations done for FI without any change in FI outside the required change.

and two constants,

1. **Step interval between measurement of FI (S_{fi}):** Number of time slots between which FI is evaluated.
2. **FI Stability Counter limit (L_{sc}):** The count C_s should reach for the algorithm to change from PFA to the improved algorithm.

If we look at the evaluation of FI every few steps, we observe that it goes through a damping process before increasing in infinitesimally small amounts. Since the number of iterations required to improve to a good FI in PFA is large, we can change the selection criteria to ensure that data sent by each user remains the same. Since the change in selection criteria should occur after the FI stabilizes, we use a FI stability counter C_s which is triggered after the change in FI is lower than 0.01 and then iterate continuously till it reaches the limit set by the constant FI Stability Counter limit (L_{sc}). Once this limit is reached, the user is selected who has the highest throughput variance. Mathematically it can be written as

$$P_{k(VPFA)} = \arg \max_k \sigma_k^2 \quad (8)$$

where σ_k^2 is the throughput variance of the k^{th} user. The pseudocode of the proposed scheduling algorithm, VPFA is given in Algorithm 1.

Algorithm 1 Improved Scheduling Algorithm (VPFA)

```
procedure SELECT_USING_PF
  Select user using pure PF
  if  $T_c \% S_{fi} == 0$  then
    evaluate FI
    if  $C_s == 0$  then
       $C_s = 1$ 
    else
      if  $LastFI - FI < 0.01$  then
        increment  $C_s$ 
      else
         $C_s = 0$ 
      end if
    end if
     $LastFI = FI$ 
  end if
  if  $C_s == L_{sc}$  then
    GOTO Select_Using_LDS
  else
    GOTO Select_Using_PF
  end if
end procedure

procedure SELECT_USING_VARIANCE
  Select user with highest variance
  GOTO Select_Using_Variance
end procedure
```

5 Simulation Results and Discussion

The simulation of the existing and proposed scheduling algorithms has been programmed using Python 2.7 [7] using the NumPy [8] and SciPy [9] libraries. The simulation parameters are given in Table 1.

Table 1: Simulation Parameters

Parameter	Value
Distance-dependent path loss	COST-231
Shadowing fading	lognormal distribution
User State	Static Positions
Shadowing standard deviation	8 dB
Total downlink TX power in traffic channels	46 dBm
Cell radius	1km
System carrier frequency	2GHz
System bandwidth	10MHz

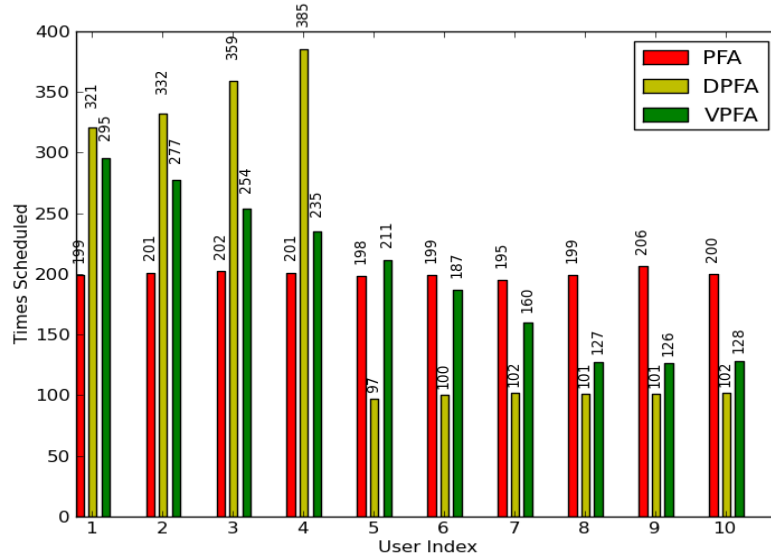


Figure 1: Comparisons of number of times individual users have been scheduled for all PF algorithms

The simulation parameters were kept constant in simulating all the algorithms so as to provide the most fair evaluation of all the algorithms. The user index increases with the decrease in distance from the tower. Figure 1 shows how the users were scheduled for all the algorithms. It is observed that the

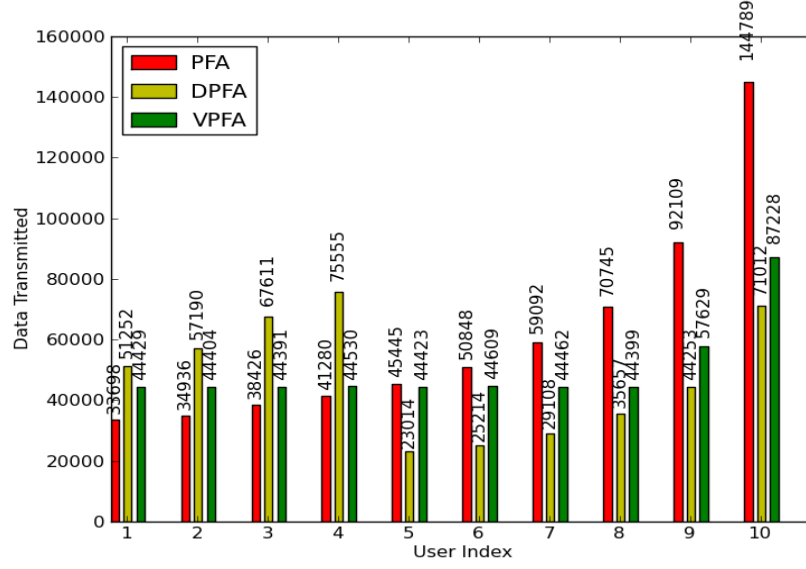


Figure 2: Comparisons of throughput of individual users for all PF algorithms

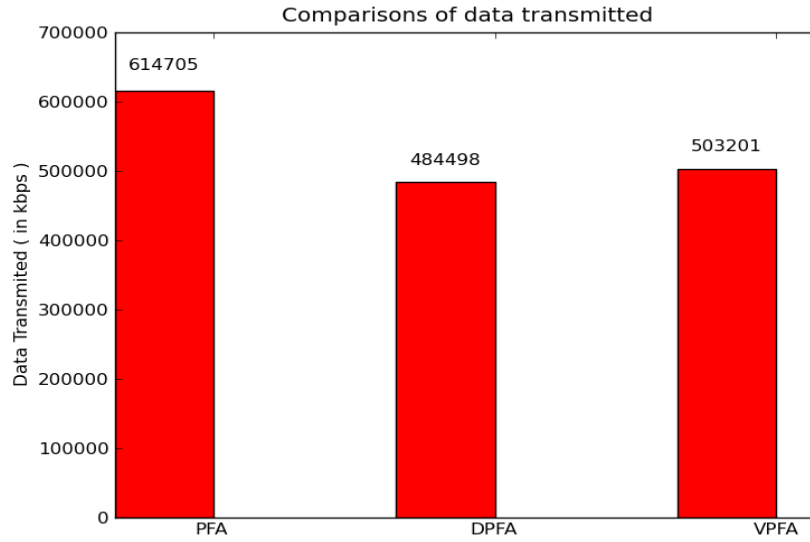


Figure 3: Comparisons of system throughput for all PF algorithms

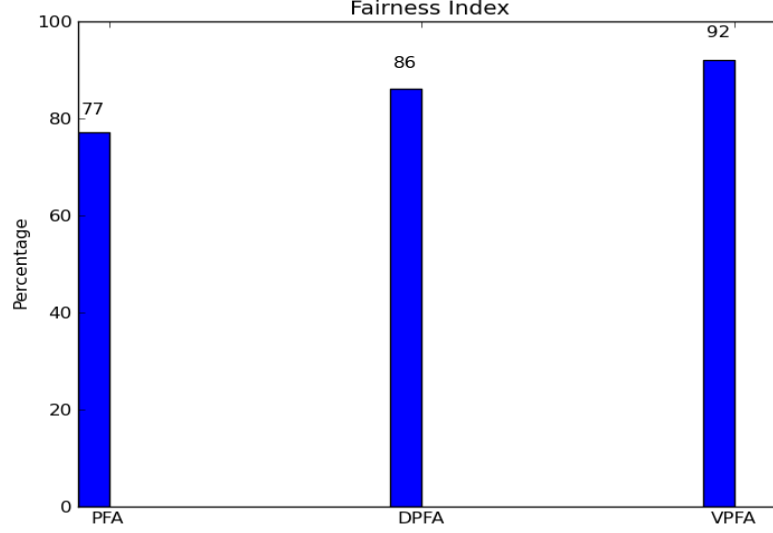


Figure 4: Comparisons of FI for all PF algorithms

users are scheduled almost equally in the PFA. DPFA schedules the cell edge users well, but punishes the cell center users for it. The proposed algorithm schedules the user most away from the tower most of the times and the number of times the user was scheduled drops as the distance from the center reduces. This causes the users to transmit almost the same amount of data as shown in Fig.2.

Considering the effect on the overall system, it is observed that even though the drop for the improved algorithm is 18% considering the PFA as reference, it is still better than the 20% drop in overall system throughput of the DPFA. Due to the fact that all the users transmit almost the same amount of data, the Fairness Index of the proposed algorithm is 0.92 which is better than the FI obtained by DPFA which is 0.86 also there is substantial improvement over PFA's 0.77. The throughput of the user at the cell edge improves by 30%. However, when the variance based selection criteria is working, the user with the smallest variance is starved. This is a drawback of the algorithm. From Fig.3, we see that the drop in overall system throughput for the VPFA is 18% as compared to DPFA's 20%. PFA is used as reference. Figure 4 shows the FI achieved by various algorithms.

6 Conclusion

In this paper by using variance of the individual user data transmitted as the selection factor in scheduling the users we are able to equalize the data transmitted by every user in the cell regardless of distance from the tower. This causes a high Fairness Index of 0.92 which is significantly higher than those obtained through PFA and DPFA. There is a appreciable improvement in the throughput of the users throughout the cell. By adopting the proposed scheduling algorithm, the QoS offered by the telecommunication companies can be improved which may lead to reduction in user attrition to other telecoms service provider hence gain in company revenue. However, the matter of starvation of the user who has the maximum throughput in the cell has scope for future research.

References

- [1] P. A. W. Harold J. Kushner, "Convergence of proportional-fair sharing algorithms under general conditions," in *IEEE Transactions on Wireless Communications*. IEEE, 2004, pp. 1250 – 1259.
- [2] X. Ning, V. Guillaume, Z. Wen, and Q. Yongquan, "A dynamic pf scheduler to improve the cell edge performance," in *Vehicular Technology Conference*, Santa Barbara, California, USA, sept 2008.
- [3] S. N. Mehrdad Dianati, Xeumin (Sherman) Shen, "A new fairness index for radio resource allocation in wireless networks," in *WCNC*. IEEE Communications Society, 2005, pp. 712–717.
- [4] R. Jain, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," <http://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf>.
- [5] X. Bi, J. Zhang, Y. Wang, and P. Viswanath, "Fairness improvement of maximum c/i scheduler by dumb antennas in slow fading channel," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, sept. 2010, pp. 1 –4.
- [6] E. Hahne, "Round-robin scheduling for max-min fairness in data networks," *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 7, pp. 1024 –1039, sep 1991.
- [7] (2012) Python 2.7. [Online]. Available: <http://www.python.org/>
- [8] (2012) Numpy. [Online]. Available: <http://numpy.scipy.org/>
- [9] (2012) Scipy. [Online]. Available: <http://www.scipy.org/>